Boundary Controlled Iterated Function Systems

Dmitry Sokolov, Gilles Gouaty, Christian Gentil, Anton Mishkinis

Boundary Controlled Iterated Function Systems

Dmitry Sokolov, Gilles Gouaty, Christian Gentil, and Anton Mishkinis

dmitry.sokolov@loria.fr
LORIA - Alice, Campus Scientifique, BP 239
54506 Vandoeuvre-les-Nancy Cedex, France

gilles_gouaty@yahoo.fr
Laboratoire des Sciences de l'Information et des Systèmes – UMR 7296
Polytech Marseille Campus de Luminy Case postale 925 13288 Marseille cedex

christian.gentil@u-bourgogne.fr
Laboratoire LE2I – UMR CNRS 6306
Faculté des Sciences Mirande, Aile de l'Ingénieur, 21078 DIJON Cedex, France

anton.mishkinis@u-bourgogne.fr
Laboratoire LE2I – UMR CNRS 6306
Faculté des Sciences Mirande, Aile de l'Ingénieur, 21078 DIJON Cedex

Abstract. Boundary Controlled Iterated Function Systems is a new layer of control over traditional (linear) IFS, allowing creation of a wide variety of shapes. In this work, we demonstrate how subdivision schemes may be generated by means of Boundary Controlled Iterated Function Systems, as well as how we may go beyond the traditional subdivision schemes to create free-form fractal shapes. BC-IFS is a powerful tool allowing creation of an object with a prescribed topology (e.g. surface patch) independent of its geometrical texture. We also show how to impose constraints on the IFS transformations to guarantee the production of smooth shapes.

Keywords: subdivision surfaces, fractals, Iterated Function System, B-splines

1 Motivation

Objects modeled through Computer Aided Geometric Design (CAGD) systems are often inspired by standard machining processes. However, other types of objects, such as objects with a porous structure or with a rough surface, may be interesting to create: porous structures can be used for their lighter weight while maintaining satisfactory mechanical properties, rough surfaces can be used for acoustic absorption.

Fractal geometry is a relatively new branch of mathematics that studies complex objects of non-integer dimensions. Because of their specific physical

properties, fractal-like structure is a centre of interest in numerous areas such as architecture [18], jewellery [19], heat and mass transport [14] or antennas [17, 6].

The emergence of techniques such as 3D printers allow for new possibilities that are as yet unused and even unexplored. Different mathematical models and algorithms have been developed to generate fractals. We can roughly categorize them into three families. The first gathers algorithms computing the basins of attraction of a given function. Julia sets and the Mandelbrot set [13] or Mandelbulb [1] are some examples. The second is based on simulation of phenomena such as percolation or diffusion [7]. The last one corresponds to deterministic or probabilistic algorithms or models based on the self-similarity property associated fractals like the terrain generator [24], Iterated Function System [3], L-system [16]. Shapes are generated from rewriting rules providing control of the geometry. Nonetheless, most of these models were developed for image synthesis without consideration of fabricability or were developed for very specific applications like Wood modeling [20].

Some studies address this aspect for specific applications for 3D printers [19]. In [5] Barnsley defines fractal homeomorphisms from $[0,1]^2$ onto the modeling space $[0,1]^2$. The same approach is used in 3D to build 3D fractals. A 3D standard object is embedded in the domain space $[0,1]^3$ and then transformed into a 3D fractal object. This approach preserves the topology of the initial object which is an important point for fabricability. The control of the resulting geometry, however, is induced by the definition of the homeomorphism and this is not obvious. And finally, the definition of the initial topology is left up to the user.

We elaborate here a new type of modeling system, using the facilities of existing CAGD software, while extending their capabilities and their application areas. This new type of modeling system will offer designers (engineers in industry) and creators (visual artists, stylists, designers, architects, etc.) new opportunities to design and produce a quick mock-up, a prototype or a single object. Our approach is to expand the possibilities of a standard CAD system by including fractal shapes while preserving ease of use for end users. We enrich Iterated Function Systems by introducing Boundary Representation concepts.

The following section introduces the necessary background information related to Iterated Function Systems and Controlled Iterated Function Systems. Section 3 presents the Boundary Controlled Iterated Function System to control the topology of fractal shapes during the subdivision process.

2 Background

2.1 Iterated Function Systems

Iterated Function Systems (IFS) were introduced by Hutchinson [10] and further developed and popularized by Barnsley [4]. More research has followed on from these seminal studies [8, 11, 3, 22].

IFS are based on the self-similarity property. A modeled object is made up of the union of several copies of itself; each copy is transformed by a function. These

functions are usually contractive, that is to say they bring points closer together and make shapes smaller. Hence, the modeled object, called the *attractor*, is made up of several possibly overlapping smaller copies of itself, each copy also made up of copies of itself, ad infinitum.

Definition. Given a complete metric space $(\mathbb{X}, d)$ with the associated metric d, an IFS is defined by a finite set of continuous transformations $T = \{T_i\}_{i=0}^{N-1}$ in the space $\mathbb{X}$. Let $\Sigma = \{0, \ldots, N-1\}$ be the set of IFS transformation indices, thus $|\Sigma| = N$. The IFS is then denoted by $\{\mathbb{X}; T_i \mid i \in \Sigma\}$.

A simple example of an IFS can be constructed for a representation of real numbers in $[0, 1]$. Let

$$T_i : x \to \frac{x+i}{3} : [0,1] \to [0,1],$$

where $i \in \Sigma = \{0, 1, 2\}$. The IFS $\{[0,1]; T_0, T_1, T_2\}$ can thus express the representation of any real number in $[0, 1]$.

We are substantially interested in so-called *hyperbolic* IFS, defined as those whose transformations T_i are all contractive.

Definition. A transformation $T : \mathbb{X} \to \mathbb{X}$ is called *contractive* if and only if there exists a real s, $0 \leqslant s < 1$ such that $d(T(x), T(y)) < s \cdot d(x, y)$ for all $x, y \in \mathbb{X}$.

Definition. For the set of non-empty compacts of $\mathbb{X}$, denoted $\mathcal{H}(\mathbb{X})$, we define the Hausdorff distance $d_{\mathbb{X}}$ induced by the metric d:

$$d_{\mathbb{X}}(A, B) = \max\{\sup_{a \in A} \inf_{b \in B} d(a, b), \sup_{b \in B} \inf_{a \in A} d(a, b)\}.$$

Since the metric space $(\mathbb{X}, d)$ is complete, the set $(\mathcal{H}(\mathbb{X}), d_{\mathbb{X}})$ is also complete.

In the early 80's, Hutchinson [10] used the Banach fixed point theorem to deduce the existence and the uniqueness of an attractor for a hyperbolic IFS, i.e. the fixed point of the associated contractive map. He defined an operator $\mathbb{T} : \mathcal{H}(\mathbb{X}) \to \mathcal{H}(\mathbb{X})$, now called Hutchinson operator [4], as the union of the IFS transformations T_i:

$$\mathbb{T}(K) = \bigcup_{i=0}^{N-1} T_i(K).$$

If IFS is hyperbolic then $\mathbb{T}$ is also contractive in the complete metric space $(\mathcal{H}(\mathbb{X}), d_{\mathbb{X}})$. According to the Banach fixed point theorem [4], $\mathbb{T}$ has a unique fixed point $\mathcal{A}$. This fixed point is named the attractor of the IFS:

$$\mathcal{A} = \mathbb{T}(\mathcal{A}) = \bigcup_{i=0}^{N-1} T_i(\mathcal{A}). \tag{1}$$

Since the Hutchinson operator is contractive, the attractor of an IFS can be evaluated recursively. That is, the attractor can be approximated by a sequence

$\{K_n\}_{n\in\mathbb{N}}$ converging to $\mathcal{A}$. The initial element in the sequence is defined by means of a primitive $K \in \mathcal{H}(\mathbb{X})$. The following elements are defined recursively:

$$\begin{aligned} K_0 &= K \\ K_{n+1} &= \bigcup_{i\in\Sigma} T_i(K_n). \end{aligned}$$

The elements K_n are images of composite functions applied to K.

Each element in the sequence represents an approximation of the IFS attractor. Each term K_n is composed of N^n images of K by a composite of n functions. For example, a sequence of the attractor approximations for an IFS $\{\mathbb{X}; T_0, T_1\}$ is presented here:

$$\begin{aligned} K_0 &= K, \\ K_1 &= \mathbb{T}(K_0) \quad = T_0(K) \cup T_1(K), \\ K_2 &= \mathbb{T}(K_1) \quad = T_0T_0(K) \cup T_0T_1(K) \cup T_1T_0(K) \cup T_1T_1(K), \\ K_3 &= \mathbb{T}(K_2) \quad = T_0T_0T_0(K) \cup T_0T_0T_1(K) \cup T_0T_1T_0(K) \cup T_0T_1T_1(K) \cup \\ &\qquad T_1T_0T_0(K) \cup T_1T_0T_1(K) \cup T_1T_1T_0(K) \cup T_1T_1T_1(K), \\ &\vdots \qquad\qquad \vdots \qquad\qquad \vdots \\ K_n &= \mathbb{T}(K_{n-1}) = \bigcup_{\alpha_i\in\{0,1\}} T_{\alpha_1}\dots T_{\alpha_n}(K_n). \end{aligned}$$

In this iterative algorithm a set of transformed primitives K is constructed recursively and calculations can be represented by an *evaluation tree*. Each node on the i-th level of the tree corresponds to the image of a composite of i IFS transformations. This tree is traversed up to a given depth n, where we display the image of K by the composite function associated with the current node, as shown in figure 1.

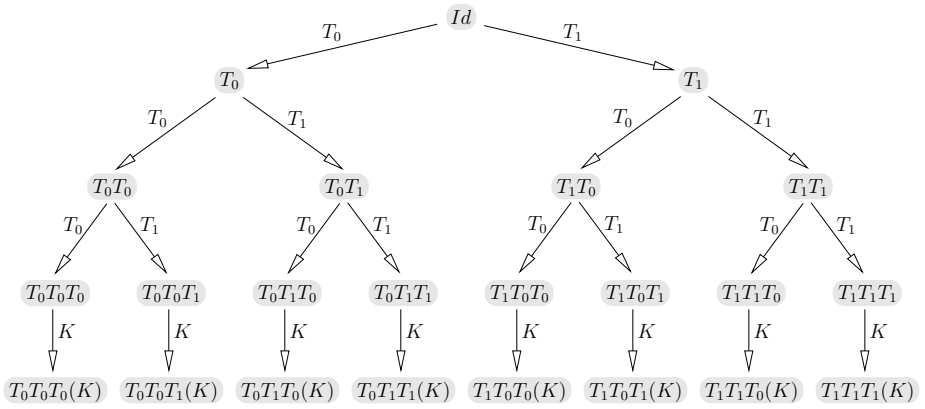

Fig. 1. The IFS evaluation tree calculated to the third level. Internal nodes correspond to the calculation of a composite function. Leaves correspond to subsets of K_3^a to construct or to visualize.

Note that these composite functions are calculated from left to right. The primitive K is transformed finally by a constructed composite function. In practice, the IFS transformations T_i are affine operators and can therefore be represented by matrices. A composite affine transformation can thus be represented by a product of transformation matrices.

2.2 Controlled / Language Restricted IFS

In IFS all the transformations are applied on each iteration. It is possible to enrich this model by adding rules to control the iterations. This is the principle of a *CIFS* (Controlled IFS).

CIFS are more general systems allowing us to control certain parts of the IFS attractor. A CIFS denotes an IFS with restrictions on transformation sequences imposed by a control graph. This system is similar to "Recurrent IFS" (RIFS) [4], and is also described [15, 21] by means of formal languages, called LRIFS (Language-Restricted Iterated Function System). CIFS defines objects whose geometry can be complex. However CIFS attractors are more convenient and controllable for manufacturability purposes than IFS attractors.

The attractor of a CIFS can be evaluated by an automaton [12] defined on the control graph. Each validated word of the automaton corresponds to an authorized composition of transformations. Each state of the automaton corresponds to different parts of the modeled object. States are associated with construction spaces. Transitions between states indicate that one sub-part is contained in another one. It is then possible to control the attractor more precisely.

Definition. A CIFS is given by an automaton, where each state q is associated with an attractor $\mathcal{A}^q \in \mathbb{X}^q$, and each transition from q to w is associated with an operator $\mathbb{X}^w \to \mathbb{X}^q$. The following is a list of parameters describing the CIFS:

- An automaton (Σ, Q, δ), where Σ is an alphabet, Q is a set of states and δ is a transition function $\delta : Q \times \Sigma \to Q$;
- A set of complete metric spaces associated with the automaton states $\{\mathbb{X}^q\}_{q\in Q}$;
- An operator associated with each transition $T_i^q : \mathbb{X}^{\delta(q,i)} \to \mathbb{X}^q$;
- A compact set $K^q \in \mathcal{H}(\mathbb{X}^q)$, called a *primitive*, associated with each state $q \in Q$. Primitives are not used to define the attractor, but only to approximate it;
- Finally, the automaton is provided by an initial state, noted by $\natural$, and all states are final states.

In the following, we denote by Σ^q the restriction of Σ by outgoing transitions from the state q, i.e.:

$$\Sigma^q = \{i \in \Sigma,\ \delta(q, i) \in Q\}$$

CIFS defines a family of attractors associated with the states: $\{\mathcal{A}^q\}_{q\in Q}$, where $\mathcal{A}^q \in \mathcal{H}(\mathbb{X}^q)$. The attractors $\mathcal{A}^q$ are mutually defined recursively:

$$\mathcal{A}^q = \bigcup_{i\in\Sigma^q} T_i^q(\mathcal{A}^{\delta(q,i)}) \tag{2}$$

As for IFS, each CIFS attractor can be approximated by a sequence $\{K_n^q\}_{n\in\mathbb{N}}$ converging to $\mathcal{A}^q$. Each state $q \in Q$ is associated with a primitive $K^q \in \mathcal{H}(\mathbb{X}^q)$, which defines the initial element in the sequence. The following elements are mutually defined recursively:

$$\begin{aligned} K_0^q &= K^q \\ K_{n+1}^q &= \bigcup_{i\in\Sigma^q} T_i^q(K_n^{\delta(q,i)}) \end{aligned}$$

In this iterative algorithm, a set of transformed primitives K^q is recursively constructed and the calculations can also be represented by an *evaluation tree*. Each node on the i-th level of the tree corresponds to the image of a composition of i CIFS transformations, i.e. a path of length i in the automaton. This tree is traversed up to a given depth n, where we display the image of K^q by the composite function associated with the current node.

Example 1 Consider an example of the CIFS attractor, illustrated in the right-hand image in figure 2. This was introduced by Bandt and Gummelt [2]. The system is described by an automaton with two states: a and b. There are two subdividing operators from state a as well as from b. The left panel in figure 2 shows the automaton of this CIFS.

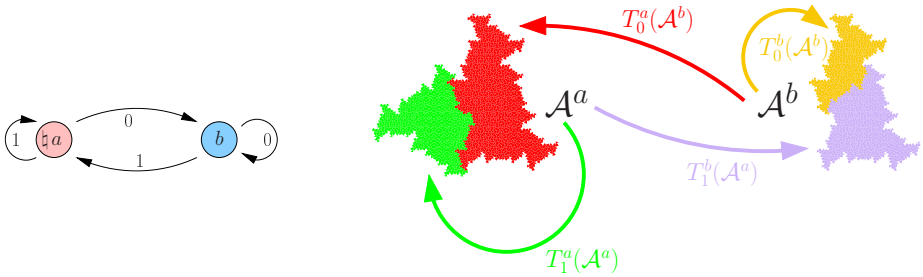

Fig. 2. Fractal kite and dart for Penrose tilings. The CIFS automaton is shown on the left, the attractors with the transformations are shown on the right.

The automaton transition functions are the following:

$$\begin{aligned} \delta(a,0) &= b & \qquad \delta(b,0) &= b \\ \delta(a,1) &= a & \qquad \delta(b,1) &= a \end{aligned}$$

Each transition $\delta(q,i) = w$ of the automaton is associated with an operator $T_i^q : \mathbb{X}^w \to \mathbb{X}^q$. **N.B.:** *$T$ and δ act in opposite directions!*

In this example, $\mathbb{X}^a$ and $\mathbb{X}^b$ are both in the same Euclidean affine plane. Let us define the mappings as follows:

$$T_0^a\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \begin{bmatrix}x\\y\end{bmatrix}$$

$$T_1^a\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \frac{2}{1+\sqrt{5}}\begin{bmatrix}\cos(3/5\pi) & -\sin(3/5\pi)\\ \sin(3/5\pi) & \cos(3/5\pi)\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}$$

$$T_0^b\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \frac{2}{1+\sqrt{5}}\begin{bmatrix}\cos(4/5\pi) & -\sin(4/5\pi)\\ \sin(4/5\pi) & \cos(4/5\pi)\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}\frac{1+\sqrt{5}}{2}\sin(4/5\pi)\\ 1+\frac{1+\sqrt{5}}{2}\cos(4/5\pi)\end{bmatrix}$$

$$T_1^b\left(\begin{bmatrix}x\\y\end{bmatrix}\right) = \frac{2}{1+\sqrt{5}}\begin{bmatrix}\cos(7/5\pi) & -\sin(7/5\pi)\\ \sin(7/5\pi) & \cos(7/5\pi)\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix} + \begin{bmatrix}0\\1\end{bmatrix}$$

Thus, the attractors $\mathcal{A}^a$ and $\mathcal{A}^b$ satisfy the following equations:

$$\mathcal{A}^a = \bigcup_{i\in\Sigma^a} T_i^a(\mathcal{A}^{\delta(a,i)}) = T_1^a(\mathcal{A}^a) \cup T_0^a(\mathcal{A}^b)$$

$$\mathcal{A}^b = \bigcup_{i\in\Sigma^b} T_i^b(\mathcal{A}^{\delta(b,i)}) = T_0^b(\mathcal{A}^b) \cup T_1^b(\mathcal{A}^a)$$

Figure 3 shows the CIFS evaluation tree calculated to the third level. Internal nodes correspond to the calculation of a composite function, while the leaves correspond to subsets of K_3^a to construct or to visualize. The pink and blue highlights help distinguish between current state (space) a and b, respectively.

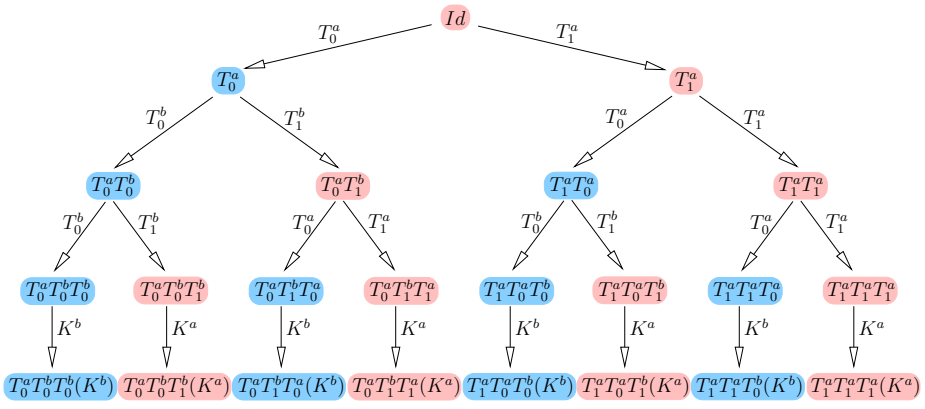

Fig. 3. CIFS evaluation tree calculated to the third level. Internal nodes correspond to the calculation of a composite function. Leaves correspond to subsets of K_3 to construct or to visualize.

Example 2 Our second CIFS example is a simple uniform quadratic B-spline curve with 4 control points. Figure 4 gives the automaton (left) and the attrac-

tors with corresponding transformations (right). This is a special kind of CIFS, sometimes called Projected IFS in the literature [23, 9].

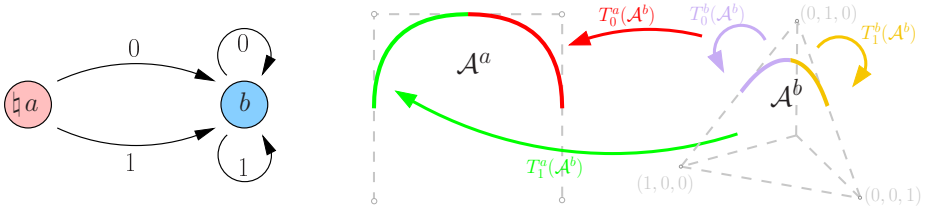

Fig. 4. The $2D$ uniform quadratic B-spline curve can be defined as a projection of an attractor from the three-dimensional barycentric space.

The automaton transition functions are the following:

$$\delta(a,0) = b \qquad \delta(b,0) = b$$
$$\delta(a,1) = b \qquad \delta(b,1) = b$$

Now the space associated with state a is still the Euclidean plane $\mathbb{R}^2$, while a three-dimensional barycentric space is associated with state b. Given the coordinates of the 4 control points P_0, P_1, P_2 and P_3, we can express the transformations as follows:

$$T_0^a\left(\begin{bmatrix}x\\y\\z\end{bmatrix}\right) = \begin{bmatrix}P_0^x & P_1^x & P_2^x\\ P_0^y & P_1^y & P_2^y\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix} \qquad T_1^a\left(\begin{bmatrix}x\\y\\z\end{bmatrix}\right) = \begin{bmatrix}P_1^x & P_2^x & P_3^x\\ P_1^y & P_2^y & P_3^y\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}$$

$$T_0^b\left(\begin{bmatrix}x\\y\\z\end{bmatrix}\right) = \begin{bmatrix}3/4 & 1/4 & 0\\ 1/4 & 3/4 & 3/4\\ 0 & 0 & 1/4\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix} \qquad T_1^b\left(\begin{bmatrix}x\\y\\z\end{bmatrix}\right) = \begin{bmatrix}1/4 & 0 & 0\\ 3/4 & 3/4 & 1/4\\ 0 & 1/4 & 3/4\end{bmatrix}\begin{bmatrix}x\\y\\z\end{bmatrix}$$

The transformations T_0^a and T_1^b are two projections of the same attractor representing the basis functions of the uniform quadratic B-spline illustrated in figure 4 (on the left side). Figure 5 gives the evaluation tree for the approximation K_3^a. Note that besides the 1st level of subdivision this is an ordinary IFS (all nodes are blue). The attractors of a CIFS are uniquely defined if operators associated with each cycle in the control graph are contractive. Hence, we do not have any constraints on the coordinates of the control points, as T_i^a do not appear in any cycle.

3 Boundary Controlled IFS

IFS and CIFS can model complex shapes; it is, however, difficult to control their topological properties. These shapes are determined by a set of geometry operators. Modifying these operators leads to both global and local changes in

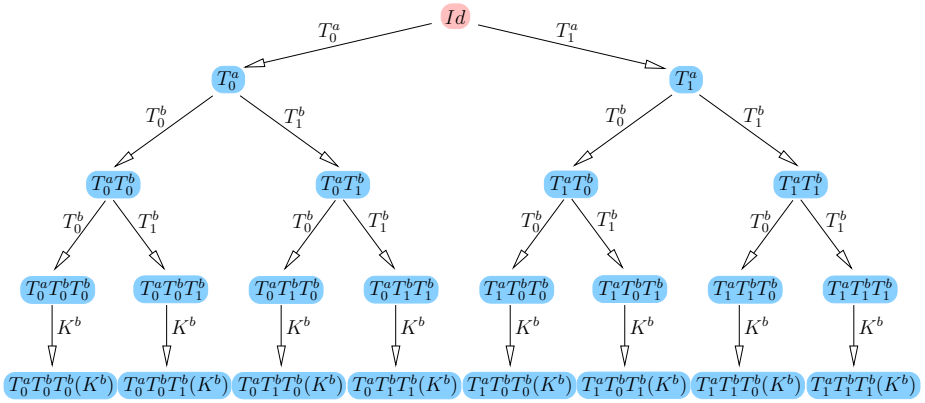

Fig. 5. CIFS evaluation tree calculated to the third level. Note that the pattern of pink/blue nodes has changed completely from the previous example.

the shape and affects not only geometry but also topology. In order to control the topological structure of the modeled shape, we enrich CIFS by integrating a topological model to obtain *BCIFS* (Boundary Controlled Iterated Function System).

In standard CAD systems, topology and geometric properties of shapes are separated. The topological structure is encoded by a set of topological cells (faces, edges, vertices) interconnected by a set of incidence and adjacency relations. The incidence relations are based on the nesting of cells: each face is bounded by a set of edges, and each edge is bounded by two vertices. The adjacency relations are based on sharing cells: two adjacent faces share a common edge, and two adjacent edges are bounded by a common vertex.

Inspired by this approach, we propose to extend the CIFS model by integrating B-rep relations. BCIFS is thus an extension of a CIFS enriched by a description of topology. Sub-parts of the attractor are identified as topological cells by specifying incidence constraints. These cells are assembled during the subdivision process by adjacency constraints. These constraints induce constraints on the subdivision operators of the CIFS.

Our B-rep structure is more general than the standard one. A topological cell may be fractal. For example, a face can be the Sierpinski triangle or an edge can be the Cantor set, but the topological structure remains consistent. Each topological cell corresponds to an attractor in a certain space.

3.1 Specifying the topology

There are two types of transitions in the BCIFS automaton:

- transitions subdividing a topological cell;
- transitions embedding a topological cell in another one.

The alphabet Σ is also divided into:

- symbols of subdivision $\Sigma_{\div} = \{\div_i \mid i = 0, \cdots, n_{\div}\}$;
- symbols of incidence $\Sigma_{\partial} = \{\partial_i \mid i = 0, \cdots, n_{\partial}\}$.

Each subdividing transition $\delta(q, \div_i) = w$ is associated with a subdividing operator $T_i^q : \mathbb{X}^w \to \mathbb{X}^q$, where $q, w \in Q$ and $\div_i \in \Sigma_{\div}$. Similarly, each embedding transition $\delta(q, \partial_i) = w$ is associated with an embedding operator $B_i^q : \mathbb{X}^w \to \mathbb{X}^q$, where $q, w \in Q$ and $\partial_i \in \Sigma_{\partial}$.

Example Let us illustrate the idea with an example. In this section we generate a continuous patch of a freeform surface with 9 control points. This patch will be defined as the attractor $\mathcal{A}^f$ of the IFS $(\mathbb{X}^f; T_0^f, T_1^f, T_2^f, T_3^f)$, let us call it "facet". We construct it as a B-rep structure with "edges" corresponding to the attractor $\mathcal{A}^e$ of the IFS $(\mathbb{X}^e; T_0^e, T_1^e)$ and "vertices" corresponding to the attractor $\mathcal{A}^v$ of the IFS $(\mathbb{X}^v; T_0^v)$.

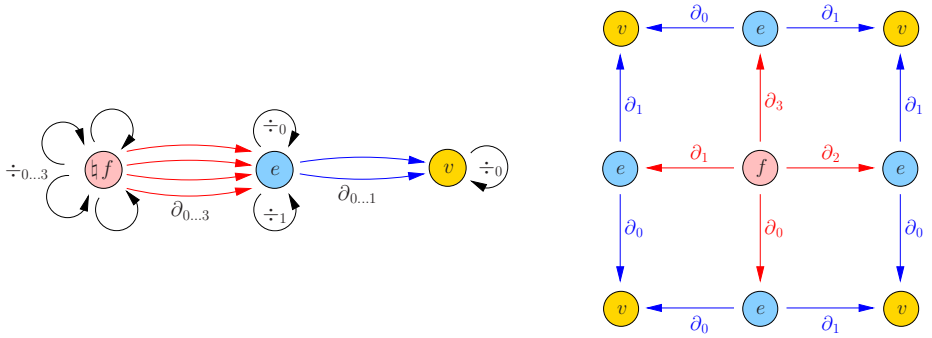

Fig. 6. Left image: automaton representing a quad patch defined by its boundaries. Right image: expanded incidence relations of the automaton .

Figure 6 gives the corresponding automaton. Note that the automaton has a hierarchical structure: there are three separate IFS linked by incidence operators. We omit the evident step of projecting the attractors into the modeling space. Refer to figure 4 to see how the projection is carried out in general.

Incidence constraints We start with the definition of the vertex attractor $\mathcal{A}^v$. To keep the example simple, we choose $\mathbb{X}^v$ to be a 1-dimensional barycentric space and T_0^v is simply a 1×1 identity matrix. The edge attractor will be defined in a 3-dimensional barycentric space.

We choose the inclusion of $\mathcal{A}^v$ (recall that it is just a point) inside the attractor $\mathcal{A}^e$, it defines boundaries of the edge $\mathcal{A}^e$. Let us say we want the vertex to be included twice at the coordinates $(1, 0, 0)$ and $(0, 0, 1)$. That is to say, we

need certain constraints on the matrices T_0^e and T_1^e to force points $(1,0,0)$ and $(0,0,1)$ to belong to the attractor $\mathcal{A}^e$.

Let us assume $B_0^e = \begin{bmatrix}1\ 0\ 0\end{bmatrix}^\top$ and $B_1^e = \begin{bmatrix}0\ 0\ 1\end{bmatrix}^\top$. Now we can express first incidence constraints as

$$B_0^e T_0^v = T_0^e B_0^e$$
$$B_1^e T_0^v = T_1^e B_0^e$$

These constraints impose a structure of the matrices T_0^e and T_1^e:

$$T_0^e = \begin{bmatrix}1 & \cdot & \cdot\\ 0 & \cdot & \cdot\\ 0 & \cdot & \cdot\end{bmatrix} \qquad T_1^e = \begin{bmatrix}\cdot & \cdot & 0\\ \cdot & \cdot & 0\\ \cdot & \cdot & 1\end{bmatrix},$$

where dots stand for arbitrarily chosen reals.

The subset of $\mathcal{A}^e$ defined as an attractor of the IFS $(\mathbb{X}^e; T_0^e)$ is equal to the $B_0^e\mathcal{A}^v$, the attractor $\mathcal{A}^v$ embedded by the action of B_0^e. In the same manner, $(0,0,1)$ is the fixed point of T_1^e and thus contained in the $\mathcal{A}^e$. Note that the first constraint implies that T_0^e must have all eigenvectors of T_0^v transformed by the action of the embedding operator B_0^e.

Property. More generally, let us show that the incidence constraints force the inclusion of the boundary CIFS attractors into the corresponding cell. If a cell $\mathcal{A}^Y$ has a number of boundaries defined by attractors $\mathcal{A}^{X_i}$, then we want to show that each $\mathcal{A}^{X_i}$ (when embedded in the space associated with state Y) is a sub part of $\mathcal{A}^Y$: in other words, we want to show that the inclusion $B_i^Y\mathcal{A}^{X_i} \subset \mathcal{A}^Y$ holds.

The incidence constraints have the following expression:

$$B_i^Y T_j^{X_i} = T_{f(i,j)}^Y B_{g(i,j)}^Y,$$

with $i \in \Sigma_\partial^Y$ and $j \in \Sigma_\div^{X_i}$. The functions f and g are simply the corresponding ordering of the boundaries and subdivisions. For example, for a square patch with four boundaries, each subdivided by two operators, we have $|\Sigma_\partial^Y| \times |\Sigma_\div^{X_i}| = 4 \times 2$ constraints.

For each boundary embedding we can write the following:

$$B_i^Y\mathcal{A}^{X_i} = B_i^Y \bigcup_{j\in\Sigma_\div^{X_i}} T_j^{X_i}\mathcal{A}^{X_i} = \bigcup_{j\in\Sigma_\div^{X_i}} B_i^Y T_j^{X_i}\mathcal{A}^{X_i} = \bigcup_{j\in\Sigma_\div^{X_i}} T_{f(i,j)}^Y B_{g(i,j)}^Y\mathcal{A}^{X_i}.$$

This means that the boundary $B_i^Y\mathcal{A}^{X_i}$ can be obtained as a union of other boundaries $B_{g(i,j)}^Y\mathcal{A}^{X_i}$ under the action of the subdivision operators. We can repeat this process ad infinitum. This means that by restricting the generated language, every boundary $B_i^Y\mathcal{A}^{X_i}$ can be generated solely by operators T^Y and therefore the inclusion $B_i^Y\mathcal{A}^{X_i} \subset \mathcal{A}^Y$ holds.

We can choose any real values for the dots in the expression of T_0^e and T_1^e, the attractor $\mathcal{A}^e$ will include two points $(1,0,0)$ and $(0,0,1)$. Note that at this point the attractor $\mathcal{A}^e$ can be a disjoint set of points. In the following subsection we add an adjacency constraint that will enable the attractor $\mathcal{A}^e$ to be a continuous curve. Figure 7 provides an example of the attractor $\mathcal{A}^e$ with the subdivision operators chosen as follows:

$$T_0^e = \begin{bmatrix} 1 & 1/2 & 1/4 \\ 0 & 1/2 & 1/2 \\ 0 & 0 & 1/4 \end{bmatrix} \qquad T_1^e = \begin{bmatrix} 1/2 & 0 & 0 \\ 1/4 & 1/2 & 0 \\ 1/4 & 1/2 & 1 \end{bmatrix},$$

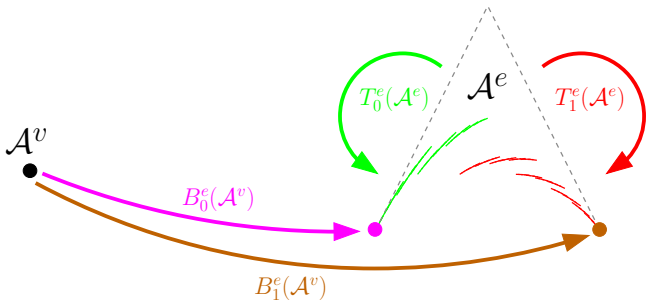

Fig. 7. Incidence constraints ensure the inclusion $B_0^e\mathcal{A}^v \subset \mathcal{A}^e$ and $B_1^e\mathcal{A}^v \subset \mathcal{A}^e$, however they do not guarantee the connectivity of the attractor $\mathcal{A}^e$.

Let us define edge-to-facet embedding operators:

$$B_0^f = \begin{bmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0 \end{bmatrix}^\top \qquad B_1^f = \begin{bmatrix} 1&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0 \end{bmatrix}^\top$$

$$B_2^f = \begin{bmatrix} 0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1 \end{bmatrix}^\top \qquad B_3^f = \begin{bmatrix} 0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&1 \end{bmatrix}^\top$$

and the corresponding incidence constraints:

$$\begin{aligned} B_0^f T_0^e &= T_0^f B_0^f & B_0^f T_1^e &= T_1^f B_0^f \\ B_3^f T_0^e &= T_2^f B_3^f & B_3^f T_1^e &= T_3^f B_3^f \\ B_1^f T_0^e &= T_0^f B_1^f & B_1^f T_1^e &= T_2^f B_1^f \\ B_2^f T_0^e &= T_1^f B_2^f & B_2^f T_1^e &= T_3^f B_2^f. \end{aligned}$$

This particular form of B_0^f simply signifies that the corresponding edge depends on the first three control points (out of nine total). If we take the first pair

of constraints only, it ensures that the attractor of the IFS $(\mathbb{X}^f; T_0^f, T_1^f)$ (recall that it is a sub-attractor of $\mathcal{A}^f$) is an image of the edge $\mathcal{A}^e$ embedded by the action of B_0^f. In the same manner, three other pairs of constraints ensure that $\mathcal{A}^f$ contains edges $B_1^f\mathcal{A}^e$, $B_2^f\mathcal{A}^e$ and $B_3^f\mathcal{A}^e$. Figure 8 shows an example of $\mathcal{A}^f$ with randomly fixed degrees of freedom.

Adjacency constraints Here we add the adjacency constraints that enforce connection of corresponding attractors. Recall that our attractors are self-similar, so after a subdivision one smaller copy of the attractor will be adjacent to another smaller copy.

Figure 9 illustrates the idea. Attractor $\mathcal{A}^e$ is defined as a union of its subdivisions $\mathcal{A}^e = T_0^e(\mathcal{A}^e) \cup T_1^e(\mathcal{A}^e)$. Let us apply the following constraint:

$$T_0^e B_1^e = T_1^e B_0^e.$$

This signifies that $T_0^e(\mathcal{A}^e)$ and $T_1^e(\mathcal{A}^e)$ must share a common vertex thus producing a connected attractor $\mathcal{A}^e$. Let us express the corresponding matrices explicitly:

$$T_0^e = \begin{bmatrix} 1 & a_0 & b_0 \\ 0 & a_1 & b_1 \\ 0 & 1-a_0-a_1 & 1-b_0-b_1 \end{bmatrix} \qquad T_1^e = \begin{bmatrix} b_0 & c_0 & 0 \\ b_1 & c_1 & 0 \\ 1-b_0-b_1 & 1-c_0-c_1 & 1 \end{bmatrix},$$

We have 6 degrees of freedom left in the matrices; any choice of the coefficients ensures the connectivity of the attractor of the IFS $(\mathbb{X}^e; T_0^e, T_1^e)$.

In exactly the same manner, we apply the adjacency constraints for the facet subdivision operators:

$$\begin{aligned} T_0^f B_2^f &= T_1^f B_1^f \\ T_2^f B_2^f &= T_3^f B_1^f \\ T_0^f B_3^f &= T_2^f B_0^f \\ T_1^f B_3^f &= T_3^f B_0^f \end{aligned}$$

Figure 10 provides an illustration. We omit an explicit expression of $T_{0\ldots3}^f$ here, since it is cumbersome but straightforward to obtain.

At this point (after applying incidence and adjacency constraints) the attractor of the IFS $(\mathbb{X}^f; T_0^f, T_1^f, T_2^f, T_3^f)$ is guaranteed to have the topology of a quad patch. The degrees of freedom left in the operators can only affect the geometric texture.

For instance, we can fix the coefficients of $T_{0\ldots3}^f$ to produce a bi-quadratic Bézier patch (left image in figure 11). But even randomly chosen coefficients produce a continuous surface (right image in figure 11).

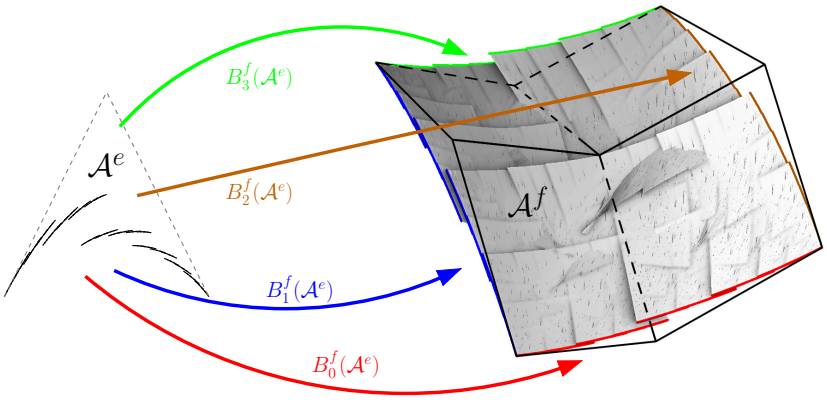

Fig. 8. As for the edge, facet incidence constraints ensure the inclusion $B_0^f\mathcal{A}^e \subset \mathcal{A}^f$, $B_1^f\mathcal{A}^e \subset \mathcal{A}^f$, $B_2^f\mathcal{A}^e \subset \mathcal{A}^f$ and $B_3^f\mathcal{A}^e \subset \mathcal{A}^f$.

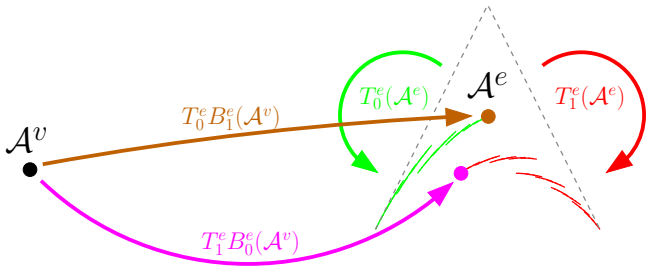

Fig. 9. In order to obtain a continuous curve, it suffices to apply the constraint $T_0^eB_1^e = T_1^eB_0^e$.

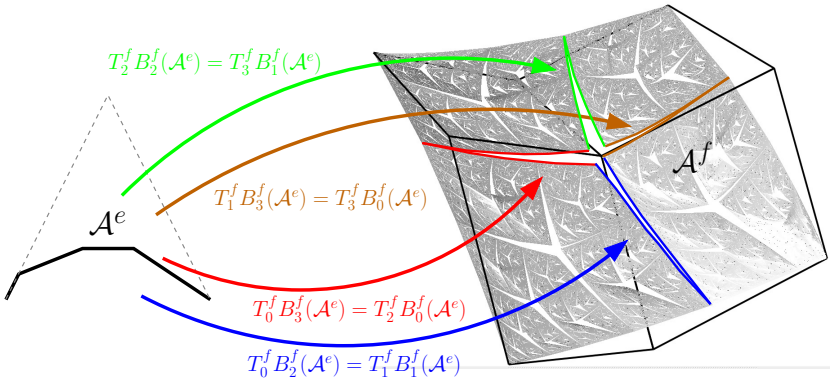

Fig. 10. Non-respect of the adjacency constraints leads to a disconnected patch.

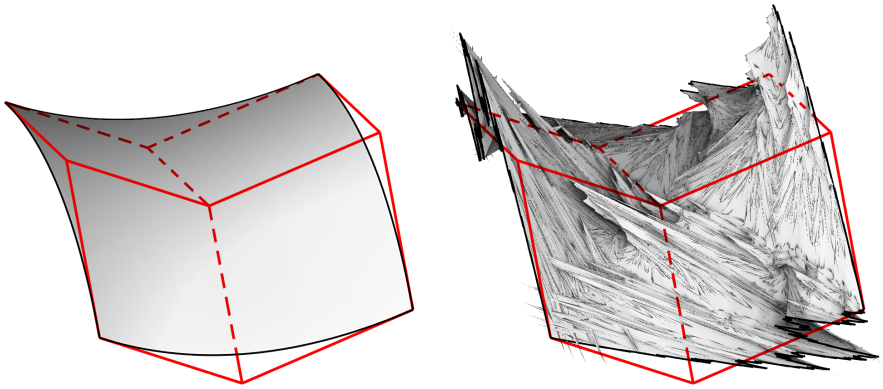

Fig. 11. After applying incidence and adjacency constraints, the attractor of the IFS $(\mathbb{X}^f; T_0^f, T_1^f, T_2^f, T_3^f)$ is guaranteed to have the topology of a quad patch. The degrees of freedom left in the operators can only affect the geometric texture. Left image: the degrees of freedom were fixed to produce a bi-quadratic Bézier patch; right image: even randomly chosen coefficients produce a continuous patch.

3.2 Controlling the geometric texture

Full analysis of the differential behaviour of produced shapes is beyond the scope of this article, but in this section we try to illustrate how the BCIFS model can be used to control geometry (in addition to topology).

Let us construct an edge bounded by two vertices, each depending on one control point. Each vertex can be represented by the same state v, the only one possible for a one dimensional space with its trivial subdivision operator : $T_0^v = \begin{bmatrix}1\end{bmatrix}$. Figure 12 provides the automaton.

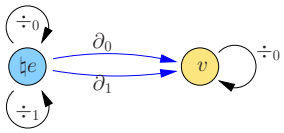

Fig. 12. In this example the edge $\mathcal{A}^e$ is bounded by two different vertices $\mathcal{A}^{v_0}$ and $\mathcal{A}^{v_1}$.

Let us assume $B_0^e = \begin{bmatrix}1\ 0\ 0\end{bmatrix}^\top$ and $B_1^e = \begin{bmatrix}0\ 0\ 1\end{bmatrix}^\top$ and the usual incidence and adjacency constraints:

$$\begin{aligned} B_0^e T_0^{v_0} &= T_0^e B_0^e \\ B_1^e T_0^{v_1} &= T_1^e B_1^e \\ T_0^e B_1^e &= T_1^e B_0^e. \end{aligned}$$

Solving for the incidence and adjacency constraints we obtain:

$$T_0^e = \begin{bmatrix} 1 \cdot a \\ 0 \cdot b \\ 0 \cdot c \end{bmatrix} \quad T_1^e = \begin{bmatrix} a \cdot 0 \\ b \cdot 0 \\ c \cdot 1 \end{bmatrix} \text{with } a+b+c=1$$

In order to control the differential behaviour at each vertex, we define two additional states, denoted by d_0 and d_1, with a two dimensional barycentric space associated with each one, and two embedding operators, $B_0^{e\prime} = \begin{bmatrix} 1\,0\,0 \\ 0\,1\,0 \end{bmatrix}^\top$ and $B_1^{e\prime} = \begin{bmatrix} 0\,1\,0 \\ 0\,0\,1 \end{bmatrix}^\top$, specifying which control points are implied for each differential behaviour. Each state has its own subdivision operator, respectively $T_0^{d_0}$ and $T_0^{d_1}$.

As for C_0 continuity, we use incidence constraints for the differential continuity.

$$B_0^{e\prime} T_0^{d_0} = T_0^e B_0^{e\prime}$$
$$B_1^{e\prime} T_0^{d_1} = T_1^e B_1^{e\prime}.$$

The consequence is:

$$T_0^{d_0} = \begin{bmatrix} 1 & 1-\lambda \\ 0 & \lambda \end{bmatrix} \qquad T_0^{d_1} = \begin{bmatrix} \mu & 0 \\ 1-\mu & 1 \end{bmatrix}$$
$$T_0^e = \begin{bmatrix} 1 & 1-\lambda & a \\ 0 & \lambda & b \\ 0 & 0 & c \end{bmatrix} \qquad T_1^e = \begin{bmatrix} a & 0 & 0 \\ b & 1-\mu & 0 \\ c & \mu & 1 \end{bmatrix}$$

Attractors of $\mathcal{A}^{d_0}$ and $\mathcal{A}^{d_1}$ are points with coordinates given by the dominant eigenvectors of $T_0^{d_0}$ and $T_0^{d_1}$. Hence we know that $\mathcal{A}^{d_0}$ is a point with coordinates $(1,0)$ and $\mathcal{A}^{d_1}$ is a point with coordinates $(0,1)$ in corresponding barycentric spaces.

Recall that incidence constraints embed all eigenvectors (and eigevalues) of $T_0^{d_0}$ and $T_0^{d_1}$ into T_0^e and T_1^e. Therefore, if λ is a sub-dominant eigenvalue of T_0^e, then the half-tangent at the "left" endpoint of the curve $\mathcal{A}^e$ is the vector $(-1,1,0)$ (the first edge of the control polygon). In the same manner, if μ is subdominant in T_1^e, then the "right" half-tangent is the vector $(0,-1,1)$, the 2nd edge of the control polygon.

As for the C^0 adjacency constraint, we can apply the same constraint for the half-tangents: [1]

$$T_0^e B_1^{e\prime} \begin{bmatrix} -1 \\ 1 \end{bmatrix} = T_1^e B_0^{e\prime} \begin{bmatrix} -1 \\ 1 \end{bmatrix} \Rightarrow T_0^e \begin{bmatrix} 0 \\ -1 \\ 1 \end{bmatrix} = T_1^e \begin{bmatrix} -1 \\ 1 \\ 0 \end{bmatrix}.$$

[1] Strictly speaking, we do not need the equality of the halftangents, collinearity suffice, so the adjacency constraint can be written as $T_0^e B_1^{e\prime} \begin{bmatrix} -\alpha \\ \alpha \end{bmatrix} = T_1^e B_0^{e\prime} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ for some $\alpha \neq 0$.

Solving the incidence and adjacency constraints we obtain the following expression for the subdivision operators:

$$T_0^e = \begin{bmatrix} 1 & 1-\lambda & \frac{1-\lambda}{2} \\ 0 & \lambda & \frac{\lambda+\mu}{2} \\ 0 & 0 & \frac{1-\mu}{2} \end{bmatrix} \qquad T_1^e = \begin{bmatrix} \frac{1-\lambda}{2} & 0 & 0 \\ \frac{\lambda+\mu}{2} & \mu & 0 \\ \frac{1-\mu}{2} & 1-\mu & 1 \end{bmatrix}.$$

Provided that λ and μ are positive subdominant eigenvalues ($\lambda \geq \frac{1-\mu}{2}$ and $\mu \geq \frac{1-\lambda}{2}$), the attractor $\mathcal{A}^e$ is guaranteed to be a C^1 curve.

3.3 Example of application

Given a set of control points and a subdivision method, construction of a CIFS whose attractor is exactly the limit subdivision surface is straightforward. In this example, we push the concept a bit further. We want to construct a solid arborescent structure, whose boundary is a subdivision surface.

Figure 13 shows the core of the subdivision process. The final arborescent structure $\mathcal{A}^a$ is defined as an iterative condensation of the attractor $\mathcal{A}^b$. Here $\mathcal{A}^b$ is a limit subdivision surface for a given mesh. The idea is simple: $\mathcal{A}^a$ is defined layer-by-layer. $\mathcal{A}^b$ covers the top of the shape, then the second layer is defined by four smaller copies of $\mathcal{A}^b$, the third layer has 16 copies of $\mathcal{A}^b$ and so forth.

Both $\mathcal{A}^a$ and $\mathcal{A}^b$ have six facets; figure 14 shows the constraints we obtain on facets from the nature of the subdivision process. We are free to choose two facets (one upper and one lateral) for the attractor $\mathcal{A}^b$, the other four are fixed automatically by our choice.

Finally, figure 15 gives the final shape of the attractor $\mathcal{A}^a$.

4 Conclusion

Our goal is to provide a computer aided geometry design software to model fractal shapes with the facilities of standard systems. The model proposed here is based on Iterated Function Systems (IFS) and enriched with Boundary Representation concepts to describe fractal topology. The fractal subdivision process is controlled by introducing incidence and adjacency constraints on topological cells. These constraints induce constraints on transformations composing the IFS. The local aspect of the shapes (rough or smooth) is controlled by the remaining degrees of freedom. The global geometry of the shape is controlled by a set of control points. This model can produce curves, surfaces, volumes, trees, or any complex fractal topology. The main important characteristic of our model is the control of the topological subdivision. According to this topological description, the fractal can be easily approximated by a coherent topological standard structure such as a mesh in order to fabricate it by 3D printing.

Descriptions are quite easy to specify for curves, surfaces or wireframe structures. But for volume subdivisions, the number of incidence and adjacency constraints increases and the description could become tedious. Furthermore, adjacency constraints have to verify orientation conditions (two adjacent cells must

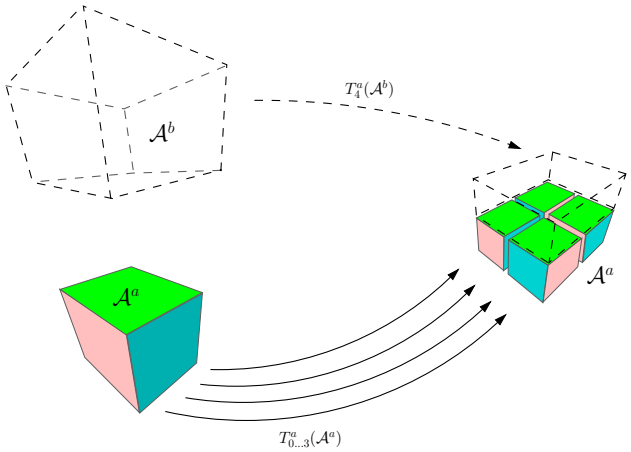

Fig. 13. Arborescent structure of $\mathcal{A}^a$ is defined by iterative condensation of the attractor $\mathcal{A}^b$.

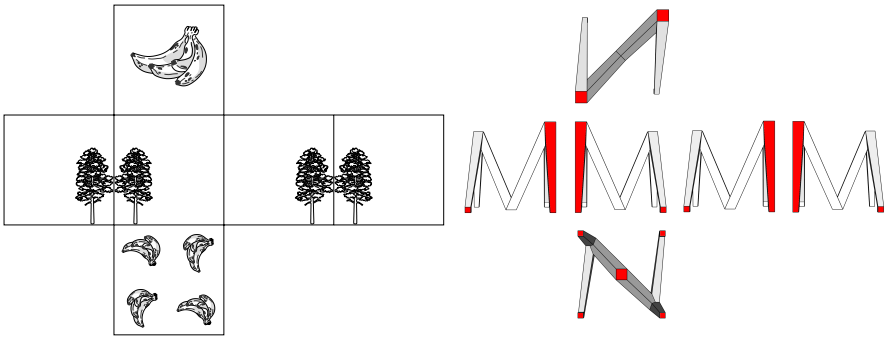

Fig. 14. Left: unfolding of the six facets of $\mathcal{A}^b$. We are free to choose two of them; the other four are fixed automatically by our choice. Right: our choice.

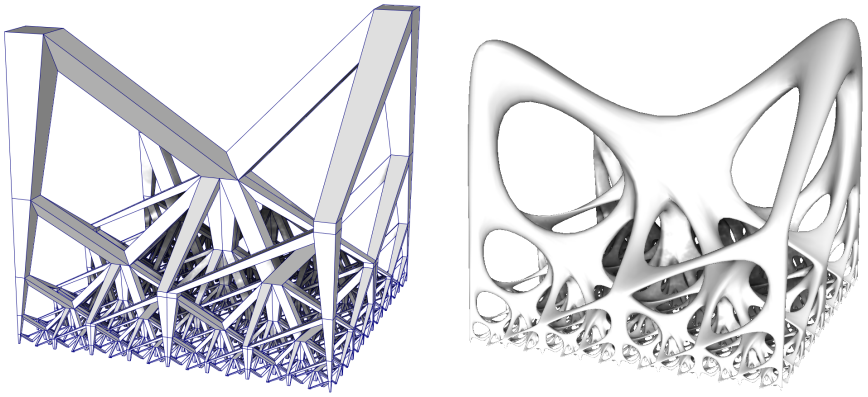

Fig. 15. Left: mesh of control points for the $\mathcal{A}^a$, right: the final shape of $\mathcal{A}^a$.

share common borders dispatched in a compatible way) to avoid degenerated solutions (attractor reduced to a point). These conditions increase the complexity of the description. However, automatic construction can be provided using topological operators as topological products.

References

1. Jacob Aron. The mandelbulb: first 'true' 3d image of famous fractal. *New Scientist*, 204(2736):54 –, 2009.
2. C. Bandt and P. Gummelt. Fractal penrose tilings i. construction and matching rules. *aequationes mathematicae*, 53(1-2):295–307, 1997.
3. M. Barnsley, J. Hutchinson, and O. Stenflo. V-variable fractals: Fractals with partial self similarity. *Advances in Mathematics*, 218(6):2051 – 2088, 2008.
4. Michael Barnsley. *Fractals everywhere*. Academic Press Professional, Inc., San Diego, CA, USA, 1988.
5. Michael Barnsley and Andrew Vince. Fractal homeomorphism for bi-affine iterated function systems. *Int. J. Applied Nonlinear Science*, 1(1):3–19, 2013.
6. N. Cohen. Fractal antenna applications in wireless telecommunications. In *Electronics Industries Forum of New England, 1997. Professional Program Proceedings*, pages 43–49, May 1997.
7. H.J. Falconer. *Fractal geometry : mathematical foundations ands applications*. Wiley, 1990. 2nd edition.
8. C. Gentil. *Les fractales en synthése d'images: le modèle IFS*. PhD thesis, Université LYON I, March 1992. Jury: D. Vandorpe, P. Chenin, J. Mazoyer, J. P. Reveilles, J. Levy Vehel, M. Terrenoire, E. Tosan.
9. E. Guerin, E. Tosan, and A Baskurt. Fractal coding of shapes based on a projected ifs model. In *Image Processing, 2000. Proceedings. 2000 International Conference on*, volume 2, pages 203–206 vol.2, Sept 2000.
10. J. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30:713 – 747, 1981.
11. P. R. Massopust. Fractal functions and their applications. *Chaos, Solitons & Fractals*, 8(2):171 – 190, 1997.
12. R. D. Mauldin and S. C. Williams. Hausdorff dimension in graph directed constructions. *Transactions of the American Mathematical Society*, 309(2):811 – 829, 1988.
13. Heinz-Otto Peitgen and Peter Richter. *The beauty of fractals : images of complex dynamical systems*. Springer-Verlag, Heidelberg, 1986.
14. Deborah Pence. The simplicity of fractal-like flow networks for effective heat and mass transport. *Experimental Thermal and Fluid Science*, 34(4):474 – 486, 2010. {ECI} International Conference on Heat Transfer and Fluid Flow in Microscale.
15. P. Prusinkiewicz and M. Hammel. Language-restricted iterated function systems, koch constructions, and l-systems. *SIGGRAPH'94 Course Notes*, 1994.
16. P. Prusinkiewicz and Aristid Lindenmayer. *The Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc., New York, NY, USA, 1990.
17. C. Puente, J. Romeu, R. Pous, X. Garcia, and F. Benitez. Fractal multiband antenna based on the sierpinski gasket. *Electronics Letters*, 32(1):1–2, Jan 1996.
18. Iasef Md Rian and Mario Sassone. Tree-inspired dendriforms and fractal-like branching structures in architecture: A brief historical overview. *Frontiers of Architectural Research*, 3(3):298 – 323, 2014.

19. S.C. Soo, K.M. Yu, and W.K. Chiu. Modeling and fabrication of artistic products based on {IFS} fractal representation. *Computer-Aided Design*, 38(7):755 – 769, 2006.
20. Olivier Terraz, Guillaume Guimberteau, Stéphane Mérillou, Dimitri Plemenos, and Djamchid Ghazanfarpour. 3gmap l-systems: an application to the modelling of wood. *The Visual Computer*, 25(2):165–180, 2009.
21. J. Thollot and E. Tosan. Construction of fractales using formal languages and matrices of attractors. In Harold P. Santos, editor, *Conference on Computational Graphics and Visualization Techniques, Compugraphics*, pages 74 – 78, Alvor, Portugal, 1993.
22. E. Tosan. Surfaces fractales définies par leurs bords. Grenoble, 2006. Journées Courbes, surfaces et algorithmes.
23. Chems Eddine Zair and Eric Tosan. Fractal modeling using free form techniques. *Computer Graphics Forum*, 15(3):269–278, 1996.
24. Howard Zhou, Jie Sun, Greg Turk, and James M. Rehg. Terrain synthesis from digital elevation models. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):834–848, July/August 2007.